ISD Preliminary Design

Number: 580-SP-041-01
Effective Date: February 18, 2006
Expiration Date: February 18, 2010

Approved By:
Name: Barbara Pfarr
Title: Assoc. Chief, ISD

Responsible Office: 580/Information Systems Division (ISD)
Title: ISD Preliminary Design

Asset Type: Sub-process
PAL Number: 2.3.1

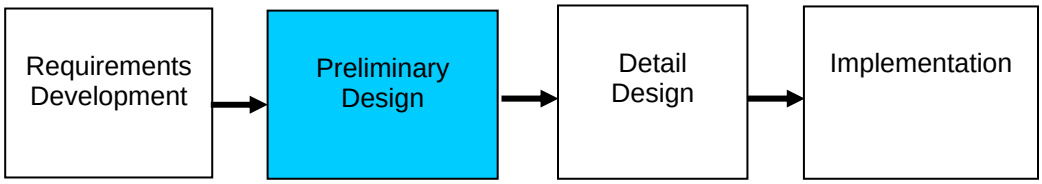
Purpose

The purpose of Preliminary Design is to provide the product capabilities and architecture for Detail Design.

Scope

This procedure is applicable to the design phase of all ISD mission software projects.

Context Diagram



Roles and Responsibilities

Product Development Lead (PDL):

Initiates the process.

Development Team Lead(s) (DTL):

Supports the PDL in executing this process.

Usage Scenarios

This process is used to establish product capabilities and architecture. This includes defining the partitions of the product, task/component/subsystem identification, internal task/component/subsystem interfaces and external interfaces.

This process may be re-entered and its products affected because of Preliminary Design Review (PDR) action items/ Review Item Dispositions (RIDs).

This process may be re-entered upon major changes to requirements – the Detailed Design process may be entered iteratively depending upon the development model but iterations should not be significant enough to change the Preliminary Design and architecture.

Inputs

Inputs include:

- Validated Requirements
- Operations Scenarios
- Interface Documents
- Re-Use Plans

- Hardware Design Architecture Constraints (including software performance requirements)
- RIDs/RFAs from reviews

Entry Criteria

This process is entered when the software requirements have been baselined or when requirements are sufficiently stable (and the schedule is sufficiently aggressive) to warrant the cost and risk of proceeding.

Exit Criteria

All outputs are completed.

Output

Outputs include:

- Preliminary design (which may include architectural diagrams; high-level module designs/package specs; state transition diagrams; analysis documentation, etc.)
- Requirements Change Requests
- PDR Presentation Materials
- Updated Requirements Traceability Matrix
- List of Identified long lead-time activities for expedited start
- Lessons Learned
- Refined estimates of system size, effort, and schedule
- RIDs/RFAs from reviews collected and placed under appropriate configuration control for tracking to closure
- Operations Concept (or updated operations scenarios)
- Updated interface documents

GUIDANCE: See the Process Asset Library (PAL) for further guidance and templates for Traceability Matrix, Preliminary Design Review and Lessons Learned.

Major Tasks

The PDL and DTL(s) shall perform sequentially, concurrently, and/or iteratively as necessary:

1. Understand Requirements
2. Design Preliminary Architecture
3. Define Interfaces
4. Estimate resource utilization and performance characteristics
5. Allocate required processing
6. Define state transitions
7. Document Software Architecture
8. Document Analysis Information
9. Identify long lead-time activities for expedited start
10. Prepare PDR materials

GUIDANCE: This is an iterative process where individual tasks may be performed concurrently and/or out of sequence.

GUIDANCE: This process develops the design sufficiently to prepare for the PDR. Any actions/changes from the PDR may result in this process being re-entered and its product affected.

GUIDANCE: This process connects with the Project Planning, Project Monitoring & Control, and Requirements and Testing processes and may

provide input to updated :Risks, CM Plan, WBS, Build Plan, QA Needs, Testing Plans, and Initial Test Cases

Task 1: **Understand Requirements**

a. Review functional requirements
GUIDANCE: Identify specific processing requirements and ensure they are understood. Follow-up with requirements author and/or appropriate subject matter expert on any disconnects or misunderstandings.

b. Review interface requirements
GUIDANCE: Identify external inputs and outputs. Follow-up on any disconnects or misunderstandings.

b. Review software safety requirements
GUIDANCE: Review any software safety requirements. If such requirements exist, develop a safety assurance strategy to minimize or eliminate the potential hazards.

c. Review relevant Integration and Test needs

d. Review relevant hardware architecture requirements

e. Review all processing and performance requirements
GUIDANCE: Identify the explicit processing required to accomplish the system requirements and any derived processing requirements. Ensure requirements are adequate to define a high level partitioning of the system and allocate all processing to partitions. In flight software, for example, typically there is a system level partition which handles tasking, scheduling, command acquisition and processing, etc. and several major sub-processes which handle major components of the required processing. In ground software, for example, there may be multiple machines/servers which handle various tasks

f. Review all IT security requirements (if applicable)
GUIDANCE: Identify components whose failure could lead to a breach of system security. If these exist, develop a security assurance strategy to ensure that the design for the identified software components minimizes or eliminates the potential for breaches of system security.

Task 2: **Design Preliminary Architecture**

a. Identify alternative solutions
b. Identify high-level reusable software from prior efforts
c. Identify make/buy decisions
d. Prepare “trade studies”
e. Select from alternatives
f. Design and document the high level architecture and components

GUIDANCE: As a major decision for the project, consider using the Decision Analysis and Resolution (DAR) process to formalize the process of preliminary architecture selection. Consider choosing evaluation criteria based on system priorities such as: optimized performance (speed and/or size), ease of use, maximum reuse, reliability and maintainability).
GUIDANCE: High level data structures, commercial applications, and algorithms may be selected at this point if their selection significantly effects the architecture of components or if their selection significantly reduces or increases risk.
GUIDANCE: The low-level reuse components are defined during the detailed design process.
GUIDANCE: Consider a self or peer review/walkthrough using the Design Inspection/Walkthrough Checklist at this point

Task 3: **Define Interfaces**

a. Define user interfaces -- Identify all required user interactions with the system. Define the mechanisms for accomplishing this interaction. Define the expected system responses to each user action.
GUIDANCE: Prototyping of user interfaces is recommended. Typically Use Cases or storyboards are developed to define user interactions.

b. Define external interfaces -- Identify all necessary interfaces with external entities other than direct users. Define the expected inputs and the expected system responses to these inputs.

c. Define internal interfaces -- Define the interfaces between processors and peripherals and among major architectural partitions.
GUIDANCE: For object-oriented design efforts, all interface mechanisms should be aligned along class boundaries so that all instances of a particular interface throughout the system are accomplished in the same manner using the same classes and operations. For structured design efforts, interface mechanisms should be aligned along module interfaces.

Task 4: **Estimate resource utilization and performance characteristics:**

a. Assess processor and memory resources -- Identify all processors in the system, their speed, internal memory capacities and all mass memory size and speed characteristics available for use by the software.
GUIDANCE: Feed back on processor requirements/constraints may be necessary and reconciliation of software requirements may occur.

b. Evaluate processing capabilities -- Use modeling or prototyping to determine system capability and system performance issues such as maximum data rates, operating system overheads, disk access speeds or other items that could affect the processing allocation.

c. Develop memory and CPU performance characteristics -- Estimate resource utilization for the system being designed. Include CPU throughput, memory utilization, and I/O channel usage. The resource usage should be evaluated against requirements and other constraints.
GUIDANCE: The flight software requirement is 50% margin at this stage

of software design. (See GSFC-STD-1000 for specifics).

GUIDANCE: This task may result in feedback to (refinement of) Operational Concepts Documents/scenarios which could result in further reconciliation of requirements.

Task 5: **Allocate required processing**

Considering execution requirements, resource reserves, design quality characteristics, and component performance capabilities, allocate the processing identified in the requirements analysis to the processors / servers / dedicated machines. This allocation consists of designating the processing to be accomplished in each processor / server / dedicated machine.

Task 6: **Define state transitions**

Identify all system states and the necessary state transitions. Define the mechanisms for accomplishing state transitions and for propagating current state information throughout the system.

Task 7: **Document Software Architecture**

Generate a representation of the software architecture up to this point.
GUIDANCE: Include a box for each processor if the software is processor specific – especially for flight software. Alternatively, include a box for each subsystem.

Typical views captured in an architectural design include the decomposition of the software subsystem into design entities, computer software configuration items (CSCI), definitions of external and internal interfaces, dependency relationships among entities and system resources, and finite state machines. (NPR 7150.2 section 3.2 "Software Design")

Task 8: **Document Analysis Information**

Additional representations from this preliminary architectural design may include as appropriate:

- Operational sequence diagrams
- Timing diagrams
- Data rate graphs
- Context Diagram(s)
- Resource requirements analysis results

Task 9 **Identify long lead-time activities for expedited start**

GUIDANCE: Hardware and tool procurements and setup, office/lab space should be considered.

Task 10: **Prepare materials**

a. Preliminary design review materials
b. Traceability Matrix updates
c. Planning Refinements/Updates

GUIDANCE: As a result of more detailed knowledge of the system to be developed, the Test Plan and Product Plan (including cost and schedule

estimates) may be refined or significantly changed.

d. Lessons Learned

GUIDANCE: Consider conducting a "self evaluation" using the Design Inspection/Walkthrough checklist, and a peer evaluation using the Design Inspection/Walkthrough checklist (in the PAL) prior to preparing PDR materials.

See the PAL for further guidance and templates for Traceability Matrix, Preliminary Design Review materials and Lessons Learned.

Measures

Recommended Measures:

- Planned and actual effort spent (cost and schedule) on the preliminary design process
- Number of change requests generated

Tools and Templates

Name	Description
FSW Design Guidelines Template	FSW/Code 582 – To be developed
Software Contents of the **Mission**-Level Preliminary Design Review	ISD/Code 580 Checklist
Contents of the Software Preliminary Design Review	ISD/580 Checklist
FSW Contents of System Preliminary Design Review	FSW/582 – To be developed
FSW Preliminary Design Review Standard	FSW/582 – To be developed
Review Item Disposition (RID) Form	ISD/580 Form – To be developed
FSB Request for Action (RFA) Form	FSW/Code 582 Template

Training

Course Name	Description
Software Project Management	Week long project management class. Course ID HQ0005

References

- **Glossary:** http://software.gsfc.nasa.gov/glossary.cfm
  Defines common terms used in ISD processes
- **Process Asset Library:** http://software.gsfc.nasa.gov/process.cfm
  Library of all ISD process descriptions
- GSFC-STD-1000 – GSFC Rules for the Design, Development, Verification, and Operation of Flight Systems ("Golden Rules"), Section 3.07 "Flight Software Margin Warnings
- Page-Jones, M., The Practical Guide to Structured Systems Design, second edition, Yourdon Press (Prentice-Hall), 1988.

- Ward, P. T. and Mellor, S. J., Structured Development for Real-Time Systems, 3 volumes, Yourdon Press, 1985, 1986.
- Yourdon, E. and Constantine, L., Structured Design, Prentice-Hall, 1979.
- Software Verification and Validation: A Practitioner's Guide by Steven R. Rakitin Artech House © 1997
- Booch, Grady, Object-Oriented Design: With Applications, Benjamin/Cummings, Redwood City, CA, 1991.
- What Makes a Good Object-Oriented Design (http://ootips.org/ood-principles.html)

Quality Management System Records

Controlled Document / Description	Record Custodian
Preliminary Design document(s)	**CMO**
Requirements Traceability Matrix	**CMO**
PDR Presentation Materials	**PDL**
Completed and signed DAR Record (Form) if used	**PDL**

Change History

Version	Date	Description of Improvements
1.0	12/23/05	Initial approved version by CCB